React Lifecycle Hands-on Lab

# Objective

Build a React app named 'blogapp' with a Posts component that:  
- Fetches posts using the JSONPlaceholder API.  
- Displays titles and body content.  
- Handles errors using lifecycle hooks.

# Step 1: Create React App

Use the following commands to create and open your React app:  
npx create-react-app blogapp  
cd blogapp  
code .

# Step 2: Create Post.js

Add the following code in src/Post.js:

class Post {  
 constructor(id, title, body) {  
 this.id = id;  
 this.title = title;  
 this.body = body;  
 }  
}  
  
export default Post;

# Step 3: Create Posts.js Component

Add the following code in src/Posts.js:

import React from "react";  
import Post from "./Post";  
  
class Posts extends React.Component {  
 constructor(props) {  
 super(props);  
 this.state = { posts: [] };  
 }  
  
 loadPosts() {  
 fetch("https://jsonplaceholder.typicode.com/posts")  
 .then(response => response.json())  
 .then(data => {  
 const postObjects = data.map(post => new Post(post.id, post.title, post.body));  
 this.setState({ posts: postObjects });  
 })  
 .catch(error => {  
 console.error("Error fetching posts:", error);  
 });  
 }  
  
 componentDidMount() {  
 this.loadPosts();  
 }  
  
 componentDidCatch(error, info) {  
 alert("An error occurred in the Posts component.");  
 console.error("Error boundary caught:", error, info);  
 }  
  
 render() {  
 return (  
 <div>  
 <h1>Posts</h1>  
 {this.state.posts.map(post => (  
 <div key={post.id}>  
 <h2>{post.title}</h2>  
 <p>{post.body}</p>  
 </div>  
 ))}  
 </div>  
 );  
 }  
}  
  
export default Posts;

# Step 4: Update App.js

Modify src/App.js to include the Posts component:

import React from "react";  
import Posts from "./Posts";  
  
function App() {  
 return (  
 <div className="App">  
 <Posts />  
 </div>  
 );  
}  
  
export default App;

# Step 5: Run the App

Run the application using:  
  
npm start

# Lifecycle Methods Used

|  |  |
| --- | --- |
| Method | Purpose |
| constructor() | Initialize component state. |
| componentDidMount() | Automatically load data after component mounts. |
| componentDidCatch() | Catch and handle rendering errors. |
| render() | Display posts in the UI. |

